APPLICATION NO. 09/826,118

TITLE OF INVENTION: Wavelet Multi-Resolution Waveforms

INVENTOR: Urbain A. von der Embse

Copy of reference:

Remez-Exchange algorithm by McClellan, Parks, and Rabiner entitled "A computer program for designing optimum FIR linear phase digital filters"

A Computer Program for Designing Optimum FIR Linear Phase Digital Filters

JAMES H. McCLELLAN, Student Member, IEEE,
THOMAS W. PARKS, Member, IEEE, and
LAWRENCE R. RABINER, Member, IEEE

Abstract—This paper presents a general-purpose computer program which is capable of designing a large class of optimum (in the minimax sense) FIR linear phase digital filters. The program has options for designing such standard filters as low-pass, high-pass, bandpass, and bandstop filters, as well as multipassband-stopband filters, differentiators, and Hilbert transformers. The program can also be used to design filters which approximate arbitrary frequency specifications which are provided by the user. The program is written in Fortran, and is carefully documented both by comments and by detailed flowcharts. The filter design algorithm is shown to be exceedingly efficient, e.g., it is capable of designing a filter with a 100-point impulse response in about 20 s.

I. Introduction

This paper presents a general algorithm for the design of a large class of finite impulse response (FIR) linear phase digital filters. Emphasis is placed on a description of how the algorithm works, and several examples are included which illustrate specific applications. A unified treatment of the theory behind this approach is available in [1].

The algorithm uses the Remez exchange method [2], [3] to design filters with minimum weighted Chebyshev error in approximating a desired ideal frequency response $D(f)$. Several authors have studied the FIR design problem for special filter types using several different algorithms [4]-[13]. The advantage of the present approach is that it combines the speed of the Remez procedure with a capability for designing a large class of general filter types. While the algorithm to be described has a special section for the more common filter types (e.g., bandpass filters with multiple bands, Hilbert transform filters, and differentiators), an arbitrary frequency response can also be approximated.

Manuscript received August 6, 1973. The work of J. H. McClellan and T. W. Parks was supported by NSF Grant GK-23697.
J. H. McClellan and T. W. Parks are with Rice University, Houston, Tex.
L. R. Rabiner is with Bell Laboratories, Murray Hill, N.J. 07974.

II. Formulation of the Approximation Problem

The frequency response of an FIR digital filter with an N-point impulse response $\{h(k)\}$ is the z-transform of the sequence evaluated on the unit circle, i.e.,

$$H(f)^1 = H(z)\Big|_{z=e^{j2\pi f}} = \sum_{k=0}^{N-1} h(k)\, e^{-j2\pi kf}. \quad (1)$$

The frequency response of a linear phase filter can be written as

$$H(f) = G(f)\, e^{j\left(\frac{L\pi}{2} - \left(\frac{N-1}{2}\right) 2\pi f\right)} \quad (2)$$

where $G(f)$ is a real valued function and $L = 0$ or 1. It is possible to show that there are exactly four cases of linear phase FIR filters to consider [1]. These four cases differ in the length of the impulse response (even or odd) and the symmetry of the impulse response [positive ($L = 0$) or negative ($L = 1$)]. By positive symmetry we mean $h(k) = h(N-1-k)$, and by negative symmetry $h(k) = -h(N-1-k)$.

In all cases, the real function $G(f)$ will be used to approximate the desired ideal magnitude specifications since the linear phase term in (2) has no effect on the magnitude response of the filter. The form of $G(f)$ depends on which of the four cases is being used. Using the appropriate symmetry relations, $G(f)$ can be expressed as follows.

Case 1: Positive symmetry, odd length:

$$G(f) = \sum_{k=0}^{n} a(k) \cos(2\pi kf) \quad (3)$$

where $n = (N-1)/2$, $a(0) = h(n)$, and $a(k) = 2h(n-k)$ for $k = 1, 2, \cdots, n$.

Case 2: Positive symmetry, even length:

$$G(f) = \sum_{k=1}^{n} b(k) \cos[2\pi(k - \tfrac{1}{2})f] \quad (4)$$

where $n = N/2$ and $b(k) = 2h(n-k)$ for $k = 1, \cdots, n$.

Case 3: Negative symmetry, odd length:

$$G(f) = \sum_{k=1}^{n} c(k) \sin(2\pi kf) \quad (5)$$

where $n = (N-1)/2$ and $c(k) = 2h(n-k)$ for $k = 1, 2, \cdots, n$ and $h(n) = 0$.

Case 4: Negative symmetry, even length:

$$G(f) = \sum_{k=1}^{n} d(k) \sin[2\pi(k - \tfrac{1}{2})f] \quad (6)$$

where $n = N/2$ and $d(k) = 2h(n-k)$ for $k = 1, \cdots, n$.

Earlier efforts at designing FIR filters concentrated on Case 1 designs, but it is now possible to combine

[1] For convenience, throughout this paper the notation $H(f)$ rather than $H(e^{j2\pi f})$ is used to denote the frequency response of the digital filter.

all four cases into one algorithm. This is accomplished by noting that $G(f)$ can be rewritten as $G(f) = Q(f)P(f)$ where $P(f)$ is a linear combination of cosine functions. Thus, results that have been worked out for Case 1 can be applied to the other three cases as well. For these purposes, it is convenient to express the summations in (4)–(6) as a sum of cosines directly. Simple manipulations of (4)–(6) yield the expressions.

Case 2:

$$\sum_{k=1}^{n} b(k) \cos\left[2\pi\left(k - \tfrac{1}{2}\right)f\right] = \cos(\pi f) \sum_{k=0}^{n-1} \tilde{b}(k) \cos(2\pi k f). \quad (7)$$

Case 3:

$$\sum_{k=1}^{n} c(k) \sin(2\pi k f) = \sin(2\pi f) \sum_{k=0}^{n-1} \tilde{c}(k) \cos(2\pi k f). \quad (8)$$

Case 4:

$$\sum_{k=1}^{n} d(k) \sin\left[2\pi\left(k - \tfrac{1}{2}\right)f\right] = \sin(\pi f) \sum_{k=0}^{n-1} \tilde{d}(k) \cos(2\pi k f) \quad (9)$$

where

$$\text{Case 2:} \begin{cases} b(1) = \tilde{b}(0) + \tfrac{1}{2}\tilde{b}(1) \\ b(k) = \tfrac{1}{2}[\tilde{b}(k-1) + \tilde{b}(k)], \quad k = 2, 3, \cdots, n-1 \\ b(n) = \tfrac{1}{2}\tilde{b}(n-1) \end{cases} \quad (10)$$

$$\text{Case 3:} \begin{cases} c(1) = \tilde{c}(0) - \tfrac{1}{2}\tilde{c}(2) \\ c(k) = \tfrac{1}{2}[\tilde{c}(k-1) - \tilde{c}(k+1)], \quad k = 2, 3, \cdots, n-2 \\ c(n-1) = \tfrac{1}{2}\tilde{c}(n-2) \\ c(n) = \tfrac{1}{2}\tilde{c}(n-1) \end{cases} \quad (11)$$

$$\text{Case 4:} \begin{cases} d(1) = \tilde{d}(0) - \tfrac{1}{2}\tilde{d}(1) \\ d(k) = \tfrac{1}{2}[\tilde{d}(k-1) - \tilde{d}(k)], \quad k = 2, 3, \cdots, n-1 \\ d(n) = \tfrac{1}{2}\tilde{d}(n-1). \end{cases} \quad (12)$$

The motivation for rewriting the four cases in a common form is that a single central computation routine (based on the Remez exchange method) can be used to calculate the best approximation in each of the four cases. This is accomplished by modifying both the desired magnitude function and the weighting function to formulate a new equivalent approximation problem.

The original approximation problem can be stated as follows: given a desired magnitude response $D(f)$ and a positive weight function $W(f)$, both continuous on a compact subset $F \subset [0, \frac{1}{2}]$ (note that the sampling rate is 1.0) and one of the four cases of linear phase filters [i.e., the forms of $G(f)$], then one wishes to minimize the maximum absolute weighted error, defined as

$$\|E(f)\| = \max_{f \in F} W(f)\,|D(f) - G(f)| \quad (13)$$

over the set of coefficients of $G(f)$.

The error function $E(f)$ can be rewritten in the form

$$E(f) = W(f)\,[D(f) - G(f)] = W(f)\,Q(f)\left[\frac{D(f)}{Q(f)} - P(f)\right] \quad (14)$$

if one is careful to omit those endpoint(s) where $Q(f) = 0$. Letting $\hat{D}(f) = D(f)/Q(f)$ and $\hat{W}(f) = W(f)\,Q(f)$, then an equivalent approximation problem would be to minimize the quantity

$$\|E(f)\| = \max_{f \in F'} \hat{W}(f)\,|\hat{D}(f) - P(f)| \quad (15)$$

by choice of the coefficients of $P(f)$. The set F is replaced by $F' = F - \{\text{endpoints where } Q(f) = 0\}$.

The net effect of this reformulation of the problem is a unification of the four cases of linear phase FIR filters from the point of view of the approximation problem. Furthermore, (15) provides a simplified viewpoint from which it is easy to see the necessary and sufficient conditions which are satisfied by the best approximation. Finally, (15) shows how to calculate this best approximation using an algorithm which can do only cosine approximations. The set of necessary and sufficient conditions for this best approximation is given in the following alternation theorem [2].

Alternation theorem: If $P(f)$ is a linear combination of r cosine functions i.e.,

$$P(f) = \sum_{k=0}^{r-1} \alpha(k) \cos 2\pi k f,$$

then a necessary and sufficient condition that $P(f)$ be the unique best weighted Chebyshev approximation to a continuous function $\hat{D}(f)$ on F' is that the weighted error function $E(f) = \hat{W}(f)\,[\hat{D}(f) - P(f)]$ exhibit *at least* $r + 1$ extremal frequencies in F'.

These extremal frequencies are a set of points $\{F_i\}$, $i = 1, 2, \cdots, r+1$ such that $F_1 < F_2 < \cdots < F_r < F_{r+1}$, with $E(F_i) = -E(F_{i+1})$, $i = 1, 2, \cdots, r$ and $|E(F_i)| = \max_{f \in F'} E(f)$.

An algorithm can now be designed to make the

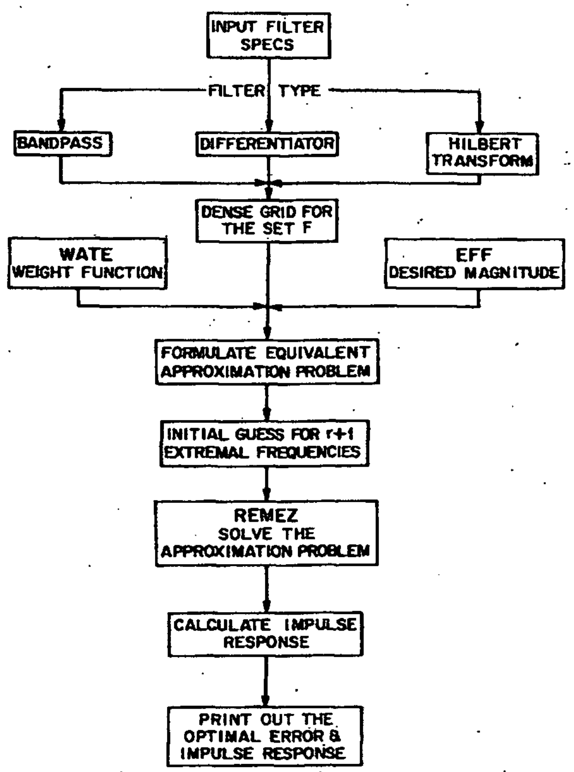

Fig. 1. Overall flowchart of filter design algorithm.

error function of the filter satisfy the set of necessary and sufficient conditions for optimality as stated in the alternation theorem. The next section describes such an algorithm along with details as to its implementation.

III. Description of the Design Algorithm

As seen in Fig. 1, the design algorithm consists of an input section, formulation of the appropriate equivalent approximation problem, solution of the approximation problem using the Remez exchange method, and calculation of the filter impulse response. The flowcharts of Figs. 2–5 give details of the exact structure of the computer program.

The input which describes the filter specifications consists of the following.

1) The filter length, $3 \leq$ NFILT $\leq$ NFMAX (the upper limit set by the programmer).

2) The type of filter (JTYPE):

a) Multiple passband/stopband (JTYPE=1)

b) Differentiator (JTYPE=2)

c) Hilbert transformer (JTYPE=3).

3) The frequency bands, specified by upper and lower cutoff frequencies (EDGE array) up to a maximum of 10 bands.

4) The desired frequency response (FX array) in each band.

5) A positive weight function (WTX array) in each band.

6) The grid density (LGRID), assumed to be 16 unless specified otherwise.

7) Impulse response punch option (JPUNCH).

Part 3) specifies the set F to be of the form $F = \cup B_i$ where each frequency band B_i is a closed subinterval of $[0, \frac{1}{2}]$. The inputs 4) and 5) are interpreted differently by the program for a differentiator than for the other two types of filters (see the EFF and WATE subroutines in Figs. 3 and 4). The weight specification in the case of a differentiator results in a *relative* error tolerance as is used in all other cases.

The set F must be replaced by a finite set of points for implementation on a computer. A dense grid of points is used with the spacing between points being $0.5/(\text{LGRID} \times r)$ where r is the number of cosine basis functions. Both $D(f)$ and $W(f)$ are evaluated on this grid by the subroutines EFF and WATE, respectively. Then the auxiliary approximation problem is set up by forming $\hat{D}(f)$ and $\hat{W}(f)$ as above, and an initial guess of the extremal frequencies is made by taking $r+1$ equally spaced frequency values. The subroutine REMEZ (Fig. 5) is called to perform the calculation of the best approximation for the equivalent problem. The mechanics of the Remez algorithm will not be discussed here since they are treated elsewhere for the particular case of low-pass filters [9]. (The flowchart of Fig. 5 gives details about the mechanics of the Remez algorithm as implemented in this design program.)

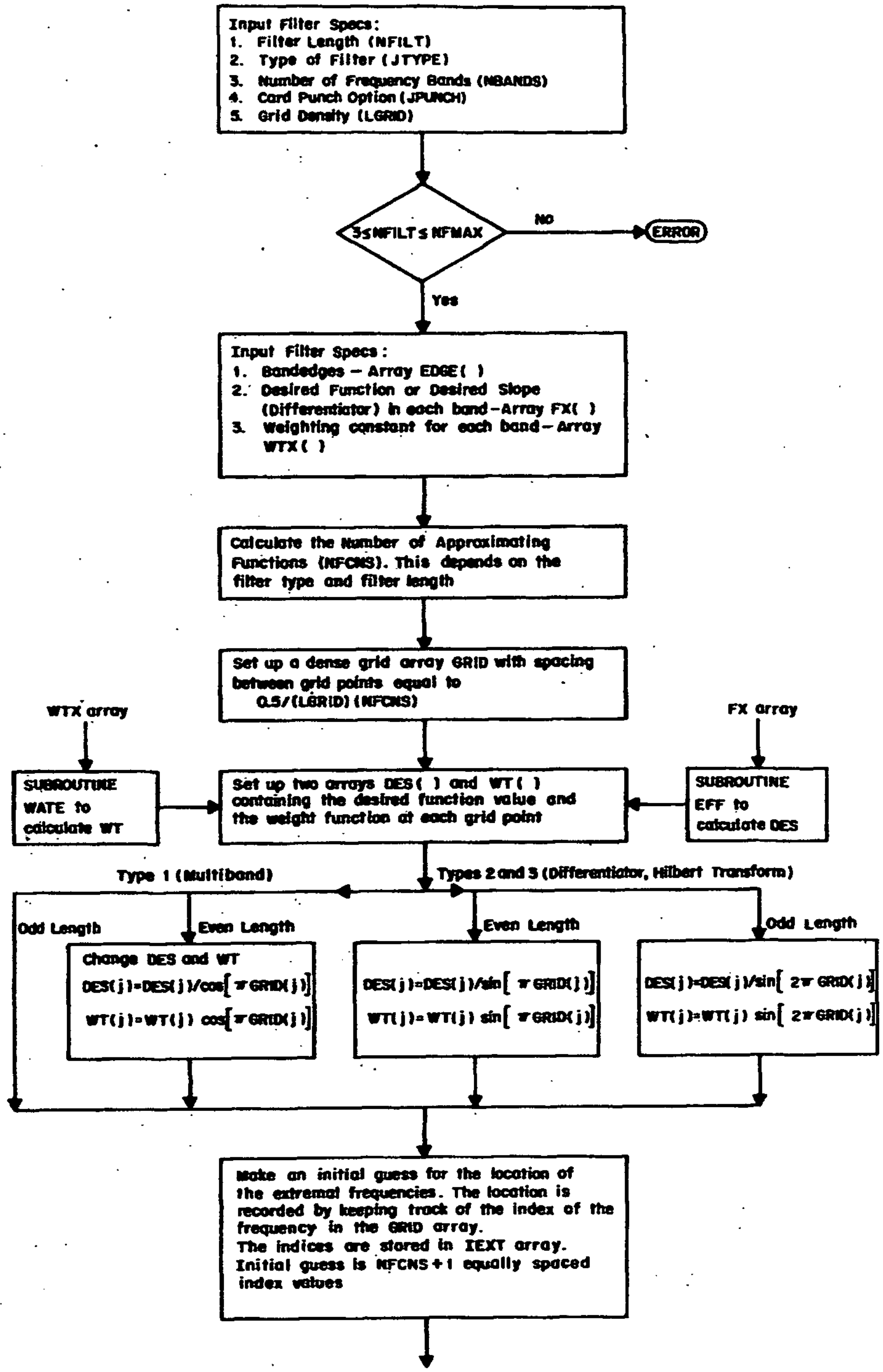

Fig. 2. Detailed flowchart for filter design algorithm.

The appropriate equations (3)–(12) are used to recover the impulse response from the coefficients of the best cosine approximation obtained in the REMEZ subroutine. The outputs of the program are the impulse response, the optimal error (min $\|E(f)\|$), and the $r + 1$ extremal frequencies where $E(f) = \pm\|E(f)\|$.

It is possible that one might want to design a filter to approximate a magnitude specification which is not included in the scheme given above, or change the weight function to get a desired tolerance scheme. A flowchart of such a program is given in Fig. 6. In such cases, the user must code the subroutines EFF and WATE to calculate $D(f)$ and $W(f)$. The input is the same as before, except that there are only two types of filters, depending on whether the impulse symmetry is positive or negative.

A detailed program listing of the generalized design program is given in the Appendix. Representative

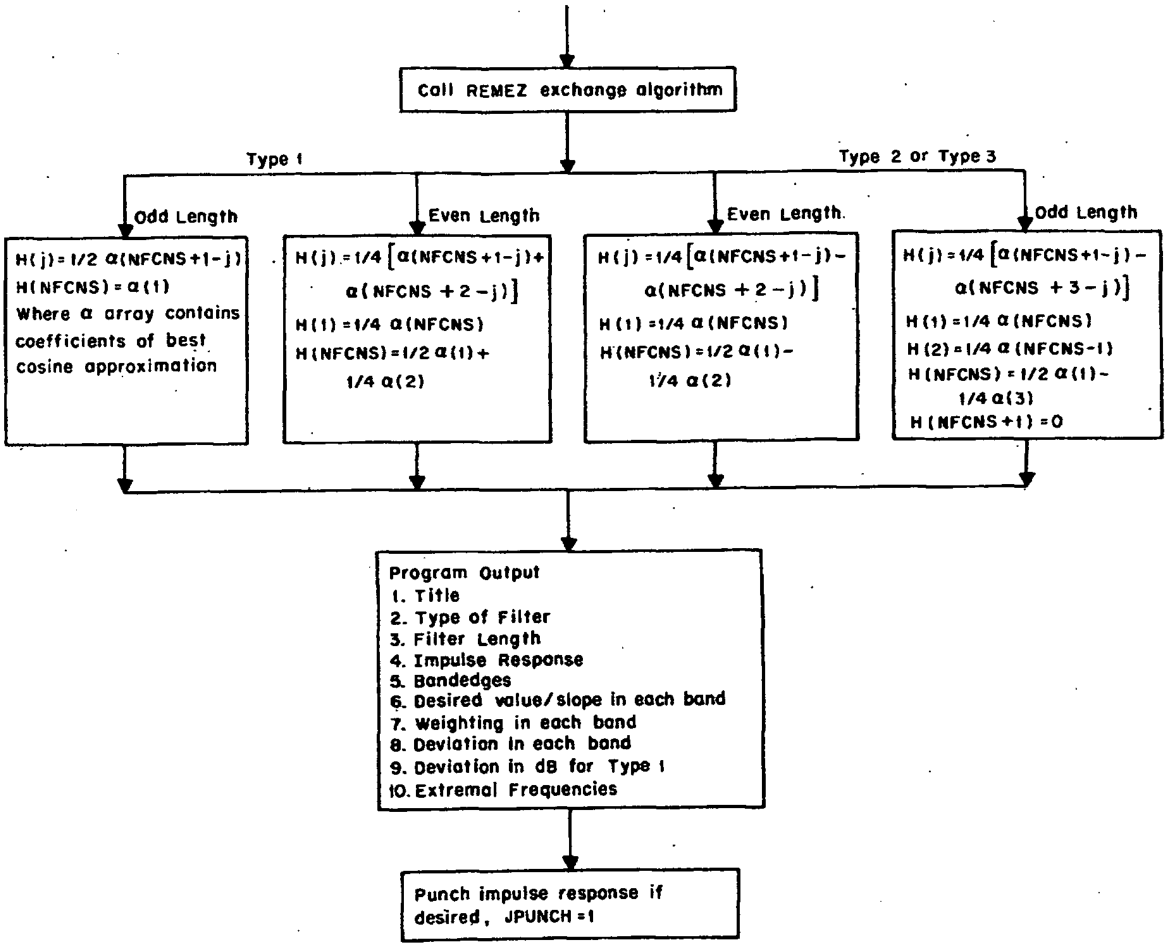

Fig. 2. (Continued.)

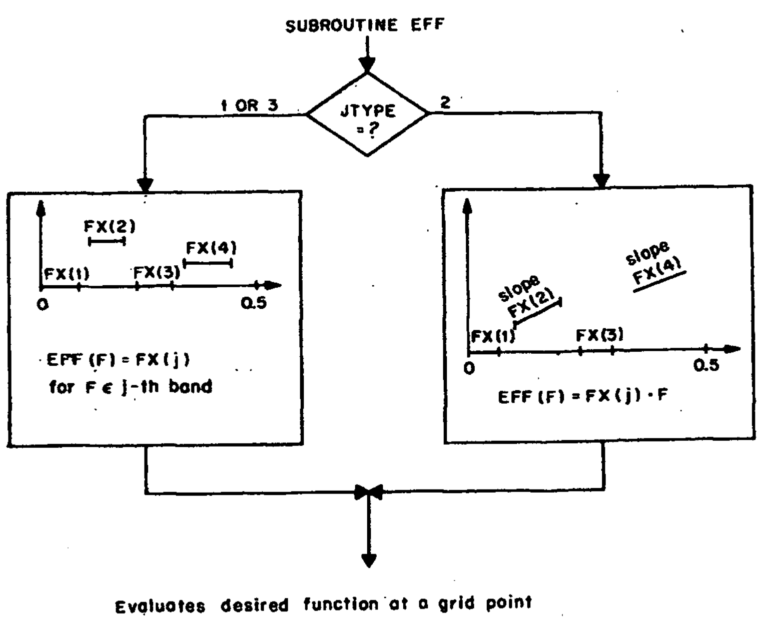

Fig. 3. Flowchart for subroutine EFF.

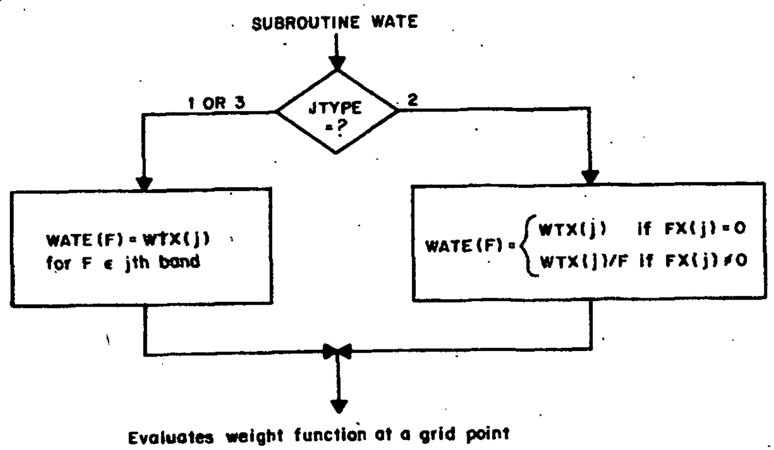

Fig. 4. Flowchart for subroutine WATE.

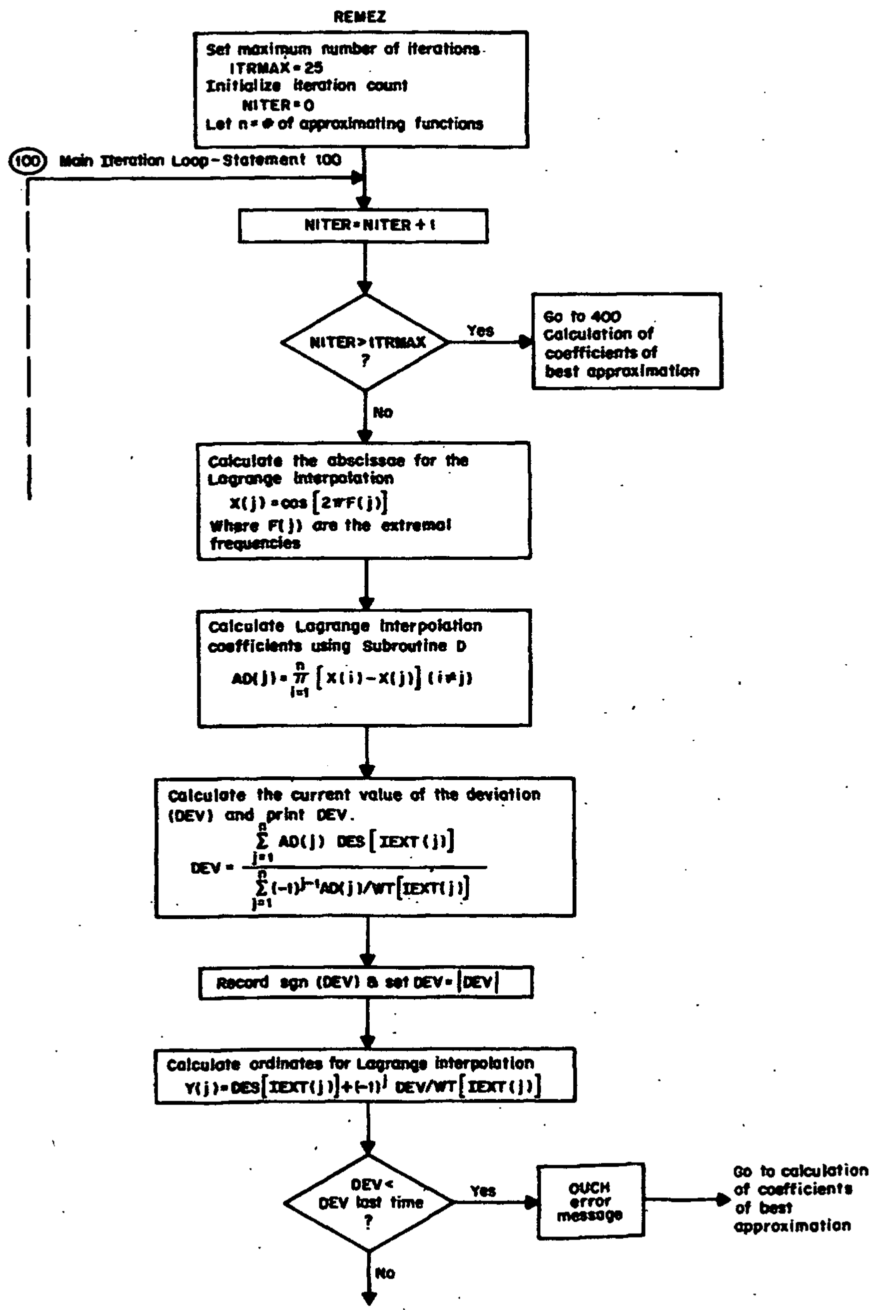

Fig. 5. Detailed flowchart for subroutine REMEZ.

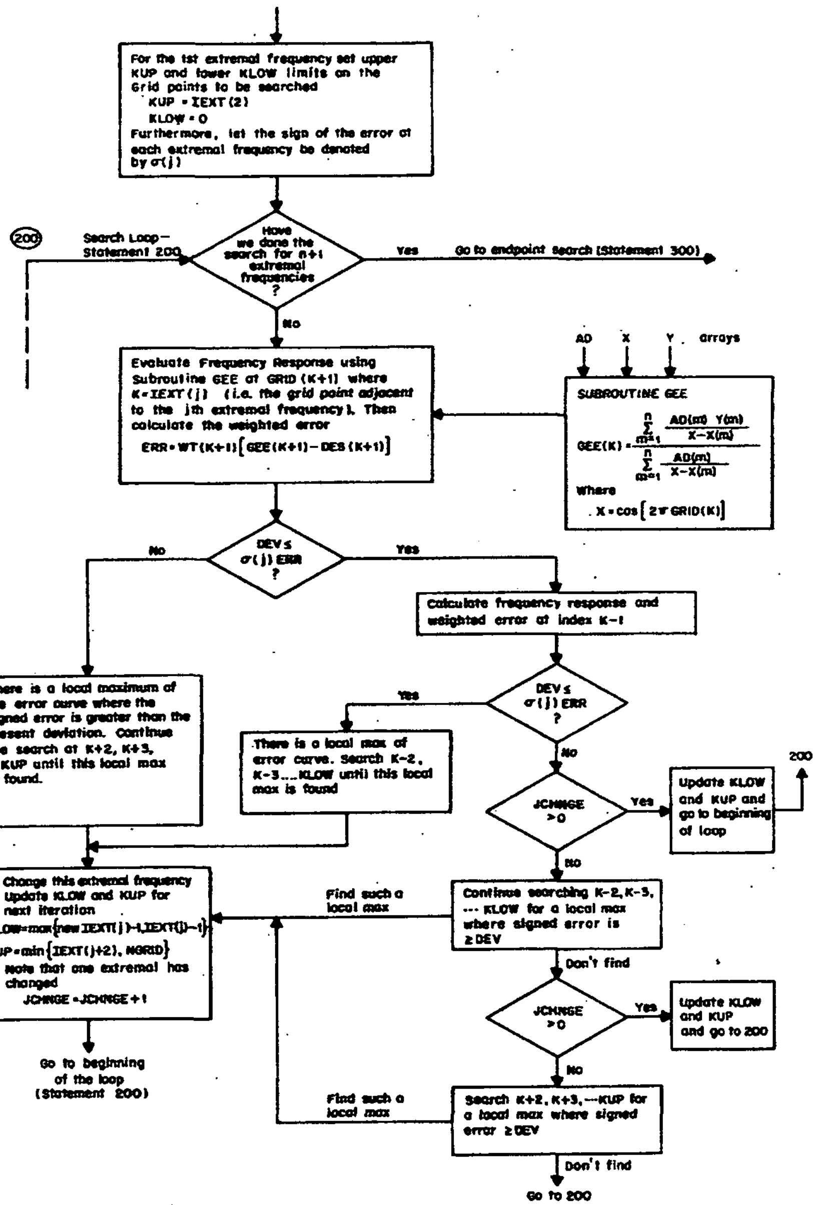

Fig. 5. (Continued.)

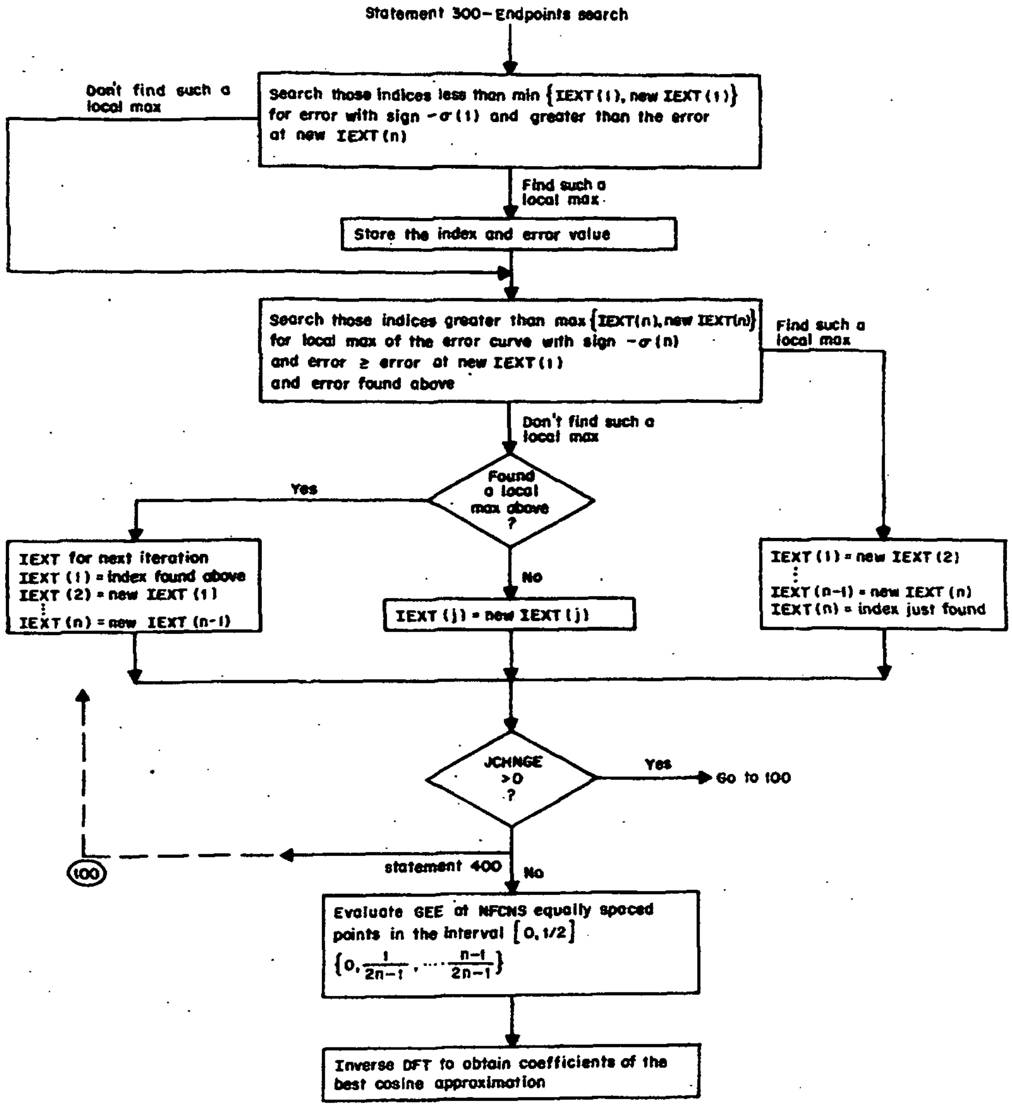

Fig. 5. (Continued.)

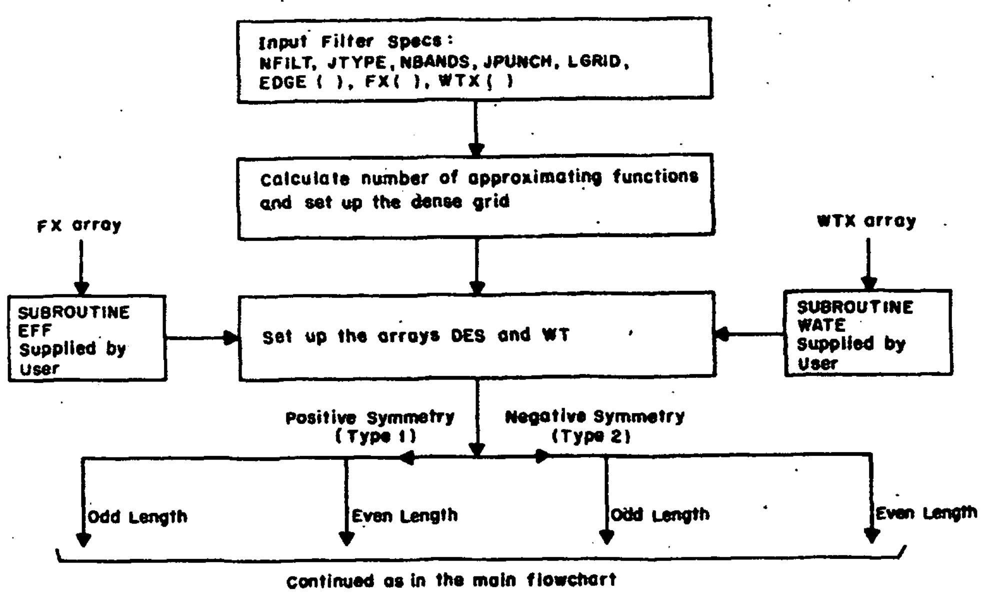

Fig. 6. Flowchart for arbitrary magnitude filter design algorithm.

```
**********************************************************************

                FINITE IMPULSE RESPONSE (FIR)
                LINEAR PHASE DIGITAL FILTER DESIGN
                REMEZ EXCHANGE ALGORITHM
                BANDPASS FILTER
           FILTER LENGTH =  24
           ***** IMPULSE RESPONSE *****
                H(  1) =  0.33740947E-02 = H(  24)
                H(  2) =  0.14938299E-01 = H(  23)
                H(  3) =  0.10569360E-01 = H(  22)
                H(  4) =  0.25415067E-02 = H(  21)
                H(  5) = -0.15929392E-01 = H(  20)
                H(  6) = -0.34085343E-01 = H(  19)
                H(  7) = -0.38112177E-01 = H(  18)
                H(  8) = -0.14629169E-01 = H(  17)
                H(  9) =  0.40089541E-01 = H(  16)
                H( 10) =  0.11540713E 00 = H(  15)
                H( 11) =  0.18850752E 00 = H(  14)
                H( 12) =  0.23354606E 00 = H(  13)

                       BAND 1           BAND 2           BAND
 LOWER BAND EDGE       0.               0.16000000
 UPPER BAND EDGE       0.08000000       0.50000000
 DESIRED VALUE         1.00000000       0.
 WEIGHTING             1.00000000       1.00000000
 DEVIATION             0.01243364       0.01243364
 DEVIATION IN DB     -38.10803413     -38.10803413

 EXTREMAL FREQUENCIES
   0.           0.0364583    0.0677083    0.0800000    0.1600000
   0.1730208    0.2008750    0.2459375    0.2870042    0.3318750
   0.3787500    0.4256251    0.4751043

**********************************************************************
TIME=       0.7694063 SECONDS
```

Fig. 7. Output listing for an $N = 24$ low-pass filter.

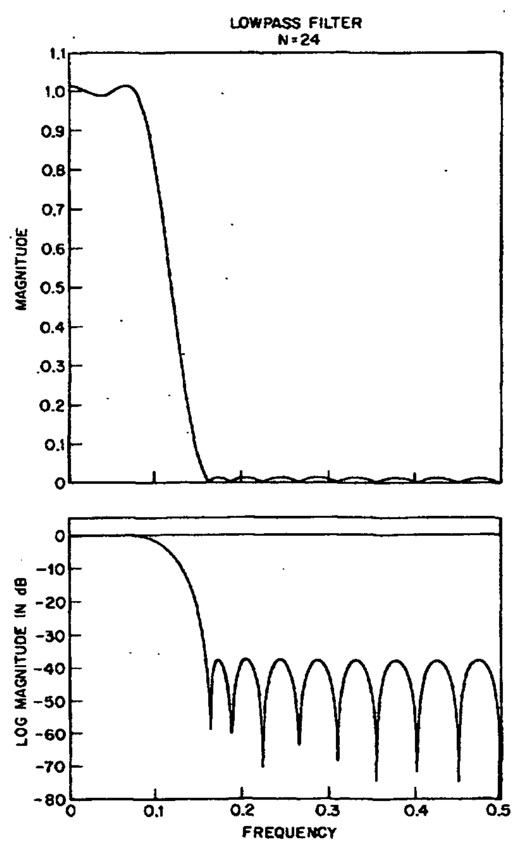

Fig. 8. Magnitude responses, on linear and log scales, for an $N = 24$ low-pass filter.

input card sequences are given for the design of a bandpass filter and a differentiator. To approximate an arbitrary magnitude response and/or an arbitrary weighting function, all the user has to do is change the subroutines EFF and WATE and use the program in the Appendix. In the next section, representative filters designed using these algorithms are presented.

IV. Design Examples

Figs. 7–22 show specific examples of use of the design program for several typical filters of interest. For each of these filters, one figure shows the computer output listing (including the run time on a Honeywell 6000 computer), and the other figure shows a plot of the filter frequency response on either a linear or a log magnitude scale (or sometimes both). Figs. 7 and 8 are for an $N = 24$ low-pass filter. For this example, the run time was 0.77 s. Figs. 9 and 10 are for an $N = 32$ bandpass filter. This example is the first example listed in the prologue to the program in the Appendix. The run time for this example was 0.82 s. Figs. 11 and 12 are for an $N = 50$ bandpass filter in which unequal weighting was used in the two stopbands. Thus the peak error in the upper stopband is ten times smaller than the peak error in the lower stopband. A total of 2.96 s was required to design this filter. Figs. 13 and 14 are for an $N = 31$ bandstop filter with equal weighting in both passbands. For the design of this filter 1.61 s were required.

To illustrate the multiband capability of the pro-

```
*************************************************************************
                FINITE IMPULSE RESPONSE (FIR)
                LINEAR PHASE DIGITAL FILTER DESIGN
                REMEZ EXCHANGE ALGORITHM
                BANDPASS FILTER
         FILTER LENGTH =  32
         ***** IMPULSE RESPONSE *****
              H(  1) = -0.57534121E-02 = H(  32)
              H(  2) =  0.99027198E-03 = H(  31)
              H(  3) = -0.75733545E-02 = H(  30)
              H(  4) = -0.65141192E-02 = H(  29)
              H(  5) =  0.13960525E-01 = H(  28)
              H(  6) =  0.22951469E-02 = H(  27)
              H(  7) = -0.19994067E-01 = H(  26)
              H(  8) =  0.71369560E-02 = H(  25)
              H(  9) = -0.39657363E-01 = H(  24)
              H( 10) =  0.11260114E-01 = H(  23)
              H( 11) =  0.66233643E-01 = H(  22)
              H( 12) = -0.10497223E-01 = H(  21)
              H( 13) =  0.85136133E-01 = H(  20)
              H( 14) = -0.12024393E 00 = H(  19)
              H( 15) = -0.29678577E 00 = H(  18)
              H( 16) =  0.30410917E 00 = H(  17)

                        BAND 1          BAND 2          BAND 3          BAND
LOWER BAND EDGE    0.               0.20000000      0.42500000
UPPER BAND EDGE    0.10000000       0.35000000      0.50000000
DESIRED VALUE      0.               1.00000000      0.
WEIGHTING         10.00000000       1.00000000     10.00000000
DEVIATION          0.00151312       0.01513118      0.00151312
DEVIATION IN DB  -56.40254641     -36.40254641    -56.40254641

EXTREMAL FREQUENCIES
   0.           0.0273437    0.0527344    0.0761719    0.0937500
   0.1000000    0.2000000    0.2195312    0.2527344    0.2839844
   0.3132812    0.3386719    0.3500000    0.4250000    0.4328125
   0.4503906    0.4796875

*************************************************************************
TIME=      0.8245625 SECONDS
```

Fig. 9. Output listing for an $N = 32$ bandpass filter.

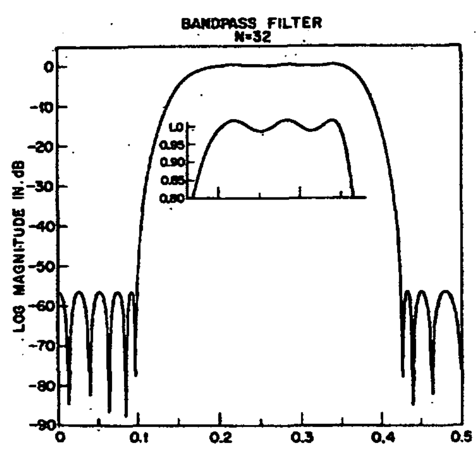

Fig. 10. Log magnitude response for an $N = 32$ bandpass filter.

```
**********************************************************************

                    FINITE IMPULSE RESPONSE (FIR)
                    LINEAR PHASE DIGITAL FILTER DESIGN
                    REMEZ EXCHANGE ALGORITHM
                    BANDPASS FILTER
          FILTER LENGTH =  55
          ***** IMPULSE RESPONSE *****
               H(  1) =  0.10662562E-02 = H(  55)
               H(  2) =  0.63777615E-02 = H(  54)
               H(  3) =  0.35755610E-02 = H(  53)
               H(  4) = -0.30677355E-02 = H(  52)
               H(  5) = -0.90306379E-02 = H(  51)
               H(  6) =  0.23155629E-02 = H(  50)
               H(  7) =  0.33637365E-02 = H(  49)
               H(  8) =  0.11172050E-01 = H(  48)
               H(  9) =  0.11646759E-01 = H(  47)
               H( 10) = -0.33630784E-02 = H(  46)
               H( 11) = -0.92384245E-02 = H(  45)
               H( 12) = -0.20406392E-01 = H(  44)
               H( 13) = -0.19460483E-01 = H(  43)
               H( 14) =  0.31243013E-01 = H(  42)
               H( 15) =  0.63045567E-02 = H(  41)
               H( 16) = -0.20482803E-01 = H(  40)
               H( 17) =  0.65740513E-02 = H(  39)
               H( 18) = -0.11202127E-02 = H(  38)
               H( 19) =  0.41956985E-01 = H(  37)
               H( 20) =  0.35784266E-01 = H(  36)
               H( 21) =  0.34744802E-01 = H(  35)
               H( 22) =  0.71496359E-01 = H(  34)
               H( 23) = -0.17138831E 00 = H(  33)
               H( 24) = -0.18255044E 00 = H(  32)
               H( 25) =  0.74059024E-01 = H(  31)
               H( 26) = -0.10317421E 00 = H(  30)
               H( 27) =  0.25716721E-01 = H(  29)
               H( 28) =  0.37813547E 00 = H(  28)

                        BAND 1          BAND 2          BAND 3          BAND 4
LOWER BAND EDGE       0.              0.10000000      0.18000000      0.30000000
UPPER BAND EDGE       0.05000000      0.15000000      0.25000000      0.36000000
DESIRED VALUE         0.              1.00000000      0.              1.00000000
WEIGHTING            10.00000000      1.00000000      3.00000000      1.00000000
DEVIATION             0.00344486      0.03444859      0.01148286      0.03444859
DEVIATION IN DB     -49.25657034    -29.25657034    -38.79839549    -29.25657034

                        BAND 5          BAND
LOWER BAND EDGE       0.41000000
UPPER BAND EDGE       0.50000000
DESIRED VALUE         0.
WEIGHTING            20.00000000
DEVIATION             0.00172243
DEVIATION IN DB     -55.27717818

EXTREMAL FREQUENCIES
   0.           0.0167411    0.0323661    0.0446429    0.0500000
   0.1000000    0.1089286    0.1267657    0.1424107    0.1500000
   0.1800000    0.1855804    0.1978571    0.2134821    0.2302232
   0.2436160    0.2500000    0.3000000    0.3122768    0.3323661
   0.3502232    0.3600000    0.4100000    0.4155804    0.4289732
   0.4457143    0.4635714    0.4814285    0.5000000

**********************************************************************
TIME=      3.8164219 SECONDS
```

Fig. 15. Output listing for an $N = 55$ multiband filter.

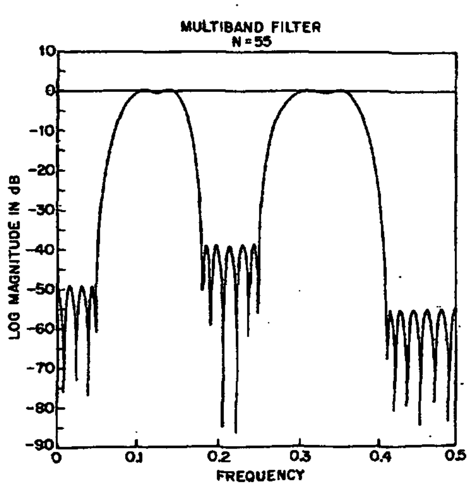

Fig. 16. Log magnitude response for an $N = 55$ multiband filter.

```
C
C  THE PROGRAM IS SET UP FOR A MAXIMUM LENGTH OF 128, BUT
C  THIS UPPER LIMIT CAN BE CHANGED BY REDIMENSIONING THE
C  ARRAYS IEXT, AD, ALPHA, X, Y, H TO BE NFMAX/2 + 2.
C  THE ARRAYS DES, GRID, AND WT MUST DIMENSIONED
C  16(NFMAX/2 + 2).
C
      NFMAX=128
  100 CONTINUE
      JTYPE=0
C
C  PROGRAM INPUT SECTION
C
      READ *,NFILT,JTYPE,NBANDS,JPUNCH,LGRID
      IF(NFILT.GT.NFMAX.OR.NFILT.LT.3) CALL ERROR
      IF(NBANDS.LE.0) NBANDS=1
C
C  GRID DENSITY IS ASSUMED TO BE 16 UNLESS SPECIFIED
C  OTHERWISE
C
      IF(LGRID.LE.0) LGRID=16
      JB=2*NBANDS
      READ *,(EDGE(J),J=1,JB)
      READ *,(FX(J),J=1,NBANDS)
      READ *,(WTX(J),J=1,NBANDS)
      IF(JTYPE.EQ.0) CALL ERROR
      NEG=1
      IF(JTYPE.EQ.1) NEG=0
      NODD=NFILT/2
      NODD=NFILT-2*NODD
      NFCNS=NFILT/2
      IF(NODD.EQ.1.AND.NEG.EQ.0) NFCNS=NFCNS+1
C
C  SET UP THE DENSE GRID.  THE NUMBER OF POINTS IN THE GRID
C  IS (FILTER LENGTH + 1)*GRID DENSITY/2
C
      GRID(1)=EDGE(1)
      DELF=LGRID*NFCNS
      DELF=0.5/DELF
      IF(NEG.EQ.0) GO TO 135
      IF(EDGE(1).LT.DELF) GRID(1)=DELF
  135 CONTINUE
      J=1
      L=1
      LBAND=1
  140 FUP=EDGE(L+1)
  145 TEMP=GRID(J)
C
C  CALCULATE THE DESIRED MAGNITUDE RESPONSE AND THE WEIGHT
C  FUNCTION ON THE GRID
C
      DES(J)=EFF(TEMP,FX,WTX,LBAND,JTYPE)
      WT(J)=WATE(TEMP,FX,WTX,LBAND,JTYPE)
      J=J+1
      GRID(J)=TEMP+DELF
      IF(GRID(J).GT.FUP) GO TO 150
      GO TO 145
  150 GRID(J-1)=FUP
      DES(J-1)=EFF(FUP,FX,WTX,LBAND,JTYPE)
      WT(J-1)=WATE(FUP,FX,WTX,LBAND,JTYPE)
      LBAND=LBAND+1
      L=L+2
      IF(LBAND.GT.NBANDS) GO TO 160
      GRID(J)=EDGE(L)
      GO TO 140
  160 NGRID=J-1
      IF(NEG.NE.NODD) GO TO 165
      IF(GRID(NGRID).GT.(0.5-DELF)) NGRID=NGRID-1
  165 CONTINUE
C
C  SET UP A NEW APPROXIMATION PROBLEM WHICH IS EQUIVALENT
C  TO THE ORIGINAL PROBLEM
C
      IF(NEG) 170,170,180
  170 IF(NODD.EQ.1) GO TO 200
      DO 175 J=1,NGRID
      CHANGE=DCOS(PI*GRID(J))
      DES(J)=DES(J)/CHANGE
  175 WT(J)=WT(J)*CHANGE
      GO TO 200
  180 IF(NODD.EQ.1) GO TO 190
      DO 185 J=1,NGRID
      CHANGE=DSIN(PI*GRID(J))
      DES(J)=DES(J)/CHANGE
  185 WT(J)=WT(J)*CHANGE
      GO TO 200
  190 DO 195 J=1,NGRID
      CHANGE=DSIN(PI2*GRID(J))
      DES(J)=DES(J)/CHANGE
  195 WT(J)=WT(J)*CHANGE
C
C  INITIAL GUESS FOR THE EXTREMAL FREQUENCIES--EQUALLY
C  SPACED ALONG THE GRID
C
  200 TEMP=FLOAT(NGRID-1)/FLOAT(NFCNS)
      DO 210 J=1,NFCNS
  210 IEXT(J)=(J-1)*TEMP+1
      IEXT(NFCNS+1)=NGRID
      NM1=NFCNS-1
      NZ=NFCNS+1
C
C  CALL THE REMEZ EXCHANGE ALGORITHM TO DO THE APPROXIMATION
C  PROBLEM
C
      CALL REMEZ(EDGE,NBANDS)
C
C  CALCULATE THE IMPULSE RESPONSE.
C
      IF(NEG) 300,300,320
  300 IF(NODD.EQ.0) GO TO 310
      DO 305 J=1,NM1
  305 H(J)=0.5*ALPHA(NZ-J)
      H(NFCNS)=ALPHA(1)
      GO TO 350
  310 H(1)=0.25*ALPHA(NFCNS)
      DO 315 J=2,NM1
  315 H(J)=0.25*(ALPHA(NZ-J)+ALPHA(NFCNS+2-J))
      H(NFCNS)=0.5*ALPHA(1)+0.25*ALPHA(2)
      GO TO 350
  320 IF(NODD.EQ.0) GO TO 330
      H(1)=0.25*ALPHA(NFCNS)
      H(2)=0.25*ALPHA(NM1)
      DO 325 J=3,NM1
  325 H(J)=0.25*(ALPHA(NZ-J)-ALPHA(NFCNS+3-J))
      H(NFCNS)=0.5*ALPHA(1)-0.25*ALPHA(3)
      H(NZ)=0.0
      GO TO 350
  330 H(1)=0.25*ALPHA(NFCNS)
      DO 335 J=2,NM1
  335 H(J)=0.25*(ALPHA(NZ-J)-ALPHA(NFCNS+2-J))
      H(NFCNS)=0.5*ALPHA(1)-0.25*ALPHA(2)
C
C  PROGRAM OUTPUT SECTION.
C
  350 PRINT 360
  360 FORMAT(1H1, 70(1H*)//25X,'FINITE IMPULSE RESPONSE (FIR)'/
     125X,'LINEAR PHASE DIGITAL FILTER DESIGN'/
     225X,'REMEZ EXCHANGE ALGORITHM'/)
      IF(JTYPE.EQ.1) PRINT 365
  365 FORMAT(25X,'BANDPASS FILTER'/)
      IF(JTYPE.EQ.2) PRINT 370
  370 FORMAT(25X,'DIFFERENTIATOR'/)
      IF(JTYPE.EQ.3) PRINT 375
  375 FORMAT(25X,'HILBERT TRANSFORMER'/)
      PRINT 378,NFILT
  378 FORMAT(15X,'FILTER LENGTH = ',I3/)
      PRINT 380
  380 FORMAT(15X,'***** IMPULSE RESPONSE *****')
      DO 381 J=1,NFCNS
      K=NFILT+1-J
      IF(NEG.EQ.0) PRINT 382,J,H(J),K
      IF(NEG.EQ.1) PRINT 383,J,H(J),K
  381 CONTINUE
  382 FORMAT(20X,'H(',I3,') = ',E15.8,' = H(',I4,')')
  383 FORMAT(20X,'H(',I3,') = ',E15.8,' = -H(',I4,')')
      IF(NEG.EQ.1.AND.NODD.EQ.1) PRINT 384,NZ
  384 FORMAT(20X,'H(',I3,') =  0.0')
      DO 450 K=1,NBANDS,4
      KUP=K+3
      IF(KUP.GT.NBANDS) KUP=NBANDS
      PRINT 385,(J,J=K,KUP)
  385 FORMAT(/24X,4('BAND',I3,8X))
      PRINT 390,(EDGE(2*J-1),J=K,KUP)
  390 FORMAT(2X,'LOWER BAND EDGE',5F15.9)
      PRINT 395,(EDGE(2*J),J=K,KUP)
  395 FORMAT(2X,'UPPER BAND EDGE',5F15.9)
      IF(JTYPE.NE.2) PRINT 400,(FX(J),J=K,KUP)
  400 FORMAT(2X,'DESIRED VALUE',2X,5F15.9)
      IF(JTYPE.EQ.2) PRINT 405,(FX(J),J=K,KUP)
  405 FORMAT(2X,'DESIRED SLOPE',2X,5F15.9)
      PRINT 410,(WTX(J),J=K,KUP)
  410 FORMAT(2X,'WEIGHTING',6X,5F15.9)
      DO 420 J=K,KUP
  420 DEVIAT(J)=DEV/WTX(J)
      PRINT 425,(DEVIAT(J),J=K,KUP)
  425 FORMAT(2X,'DEVIATION',6X,5F15.9)
      IF(JTYPE.NE.1) GO TO 450
      DO 430 J=K,KUP
  430 DEVIAT(J)=20.0*ALOG10(DEVIAT(J))
      PRINT 435,(DEVIAT(J),J=K,KUP)
  435 FORMAT(2X,'DEVIATION IN DB',5F15.9)
  450 CONTINUE
      PRINT 455,(GRID(IEXT(J)),J=1,NZ)
  455 FORMAT(/2X,'EXTREMAL FREQUENCIES'/(2X,5F12.7))
      PRINT 460
  460 FORMAT(/1X,70(1H*)/1H1)
      IF(JPUNCH.NE.0) PUNCH *,(H(J),J=1,NFCNS)
      IF(NFILT.NE.0) GO TO 100
      RETURN
      END

      FUNCTION EFF(TEMP,FX,WTX,LBAND,JTYPE)
C
C  FUNCTION TO CALCULATE THE DESIRED MAGNITUDE RESPONSE
C  AS A FUNCTION OF FREQUENCY.
C
      DIMENSION FX(5),WTX(5)
      IF(JTYPE.EQ.2) GO TO 1
      EFF=FX(LBAND)
      RETURN
```

```
    1 EFF=FX(LBAND)*TEMP
      RETURN
      END


      FUNCTION WATE(TEMP,FX,WTX,LBAND,JTYPE)
C
C  FUNCTION TO CALCULATE THE WEIGHT FUNCTION AS A FUNCTION
C  OF FREQUENCY.
C
      DIMENSION FX(5),WTX(5)
      IF(JTYPE.EQ.2) GO TO 1
      WATE=WTX(LBAND)
      RETURN
    1 IF(FX(LBAND).LT.0.0001) GO TO 2
      WATE=WTX(LBAND)/TEMP
      RETURN
    2 WATE=WTX(LBAND)
      RETURN
      END


      SUBROUTINE ERROR
      PRINT 1
    1 FORMAT(' ************ ERROR IN INPUT DATA **********')
      STOP
      END


      SUBROUTINE REMEZ(EDGE,NBANDS)
C
C  THIS SUBROUTINE IMPLEMENTS THE REMEZ EXCHANGE ALGORITHM
C  FOR THE WEIGHTED CHEBYCHEV APPROXIMATION OF A CONTINUOUS
C  FUNCTION WITH A SUM OF COSINES.  INPUTS TO THE SUBROUTINE
C  ARE A DENSE GRID WHICH REPLACES THE FREQUENCY AXIS, THE
C  DESIRED FUNCTION ON THIS GRID, THE WEIGHT FUNCTION ON THE
C  GRID, THE NUMBER OF COSINES, AND AN INITIAL GUESS OF THE
C  EXTREMAL FREQUENCIES.  THE PROGRAM MINIMIZES THE CHEBYCHEV
C  ERROR BY DETERMINING THE BEST LOCATION OF THE EXTREMAL
C  FREQUENCIES (POINTS OF MAXIMUM ERROR) AND THEN CALCULATES
C  THE COEFFICIENTS OF THE BEST APPROXIMATION.
C
      COMMON PI2,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
      DIMENSION EDGE(20)
      DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
      DIMENSION DES(1045),GRID(1045),WT(1045)
      DIMENSION A(66),P(65),Q(65)
      DOUBLE PRECISION PI2,DNUM,DDEN,DTEMP,A,P,Q
      DOUBLE PRECISION AD,DEV,X,Y
C
C  THE PROGRAM ALLOWS A MAXIMUM NUMBER OF ITERATIONS OF 25
C
      ITRMAX=25
      DEVL=-1.0
      NZ=NFCNS+1
      NZZ=NFCNS+2
      NITER=0
  100 CONTINUE
      IEXT(NZZ)=NGRID+1
      NITER=NITER+1
      IF(NITER.GT.ITRMAX) GO TO 400
      DO 110 J=1,NZ
      DTEMP=GRID(IEXT(J))
      DTEMP=DCOS(DTEMP*PI2)
  110 X(J)=DTEMP
      JET=(NFCNS-1)/15+1
      DO 120 J=1,NZ
  120 AD(J)=D(J,NZ,JET)
      DNUM=0.0
      DDEN=0.0
      K=1
      DO 130 J=1,NZ
      L=IEXT(J)
      DTEMP=AD(J)*DES(L)
      DNUM=DNUM+DTEMP
      DTEMP=K*AD(J)/WT(L)
      DDEN=DDEN+DTEMP
  130 K=-K
      DEV=DNUM/DDEN
      NU=1
      IF(DEV.GT.0.0) NU=-1
      DEV=-NU*DEV
      K=NU
      DO 140 J=1,NZ
      L=IEXT(J)
      DTEMP=K*DEV/WT(L)
      Y(J)=DES(L)+DTEMP
  140 K=-K
      IF(DEV.GE.DEVL) GO TO 150
      CALL OUCH
      GO TO 400
  150 DEVL=DEV
      JCHNGE=0
      K1=IEXT(1)
      KNZ=IEXT(NZ)
      KLOW=0
      NUT=-NU
      J=1
C
C  SEARCH FOR THE EXTREMAL FREQUENCIES OF THE BEST
C  APPROXIMATION
C
  200 IF(J.EQ.NZZ) YNZ=COMP
      IF(J.GE.NZZ) GO TO 300
      KUP=IEXT(J+1)
      L=IEXT(J)+1
      NUT=-NUT
      IF(J.EQ.2) Y1=COMP
      COMP=DEV
      IF(L.GE.KUP) GO TO 220
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 220
      COMP=NUT*ERR
  210 L=L+1
      IF(L.GE.KUP) GO TO 215
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 215
      COMP=NUT*ERR
      GO TO 210
  215 IEXT(J)=L-1
      J=J+1
      KLOW=L-1
      JCHNGE=JCHNGE+1
      GO TO 200
  220 L=L-1
  225 L=L-1
      IF(L.LE.KLOW) GO TO 250
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.GT.0.0) GO TO 230
      IF(JCHNGE.LE.0) GO TO 225
      GO TO 260
  230 COMP=NUT*ERR
  235 L=L-1
      IF(L.LE.KLOW) GO TO 240
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 240
      COMP=NUT*ERR
      GO TO 235
  240 KLOW=IEXT(J)
      IEXT(J)=L+1
      J=J+1
      JCHNGE=JCHNGE+1
      GO TO 200
  250 L=IEXT(J)+1
      IF(JCHNGE.GT.0) GO TO 215
  255 L=L+1
      IF(L.GE.KUP) GO TO 260
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 255
      COMP=NUT*ERR
      GO TO 210
  260 KLOW=IEXT(J)
      J=J+1
      GO TO 200
  300 IF(J.GT.NZZ) GO TO 320
      IF(K1.GT.IEXT(1)) K1=IEXT(1)
      IF(KNZ.LT.IEXT(NZ)) KNZ=IEXT(NZ)
      NUT1=NUT
      NUT=-NU
      L=0
      KUP=K1
      COMP=YNZ*(1.00001)
      LUCK=1
  310 L=L+1
      IF(L.GE.KUP) GO TO 315
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 310
      COMP=NUT*ERR
      J=NZZ
      GO TO 210
  315 LUCK=6
      GO TO 325
  320 IF(LUCK.GT.9) GO TO 350
      IF(COMP.GT.Y1) Y1=COMP
      K1=IEXT(NZZ)
  325 L=NGRID+1
      KLOW=KNZ
      NUT=-NUT1
      COMP=Y1*(1.00001)
  330 L=L-1
      IF(L.LE.KLOW) GO TO 340
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 330
      J=NZZ
      COMP=NUT*ERR
      LUCK=LUCK+10
      GO TO 235
```

```
  340 IF(LUCK.EQ.6) GO TO 370
      DO 345 J=1,NFCNS
  345 IEXT(NZZ-J)=IEXT(NZ-J)
      IEXT(1)=K1
      GO TO 100
  350 KN=IEXT(NZZ)
      DO 360 J=1,NFCNS
  360 IEXT(J)=IEXT(J+1)
      IEXT(NZ)=KN
      GO TO 100
  370 IF(JCHNGE.GT.0) GO TO 100


C  CALCULATION OF THE COEFFICIENTS OF THE BEST APPROXIMATION
C  USING THE INVERSE DISCRETE FOURIER TRANSFORM
C
  400 CONTINUE
      NM1=NFCNS-1
      FSH=1.0E-06
      GTEMP=GRID(1)
      X(NZZ)=-2.0
      CN=2*NFCNS-1
      DELF=1.0/CN
      L=1
      KKK=0
      IF(EDGE(1).EQ.0.0.AND.EDGE(2*NBANDS).EQ.0.5) KKK=1
      IF(NFCNS.LE.3) KKK=1
      IF(KKK.EQ.1) GO TO 405
      DTEMP=DCOS(PI2*GRID(1))
      DNUM=DCOS(PI2*GRID(NGRID))
      AA=2.0/(DTEMP-DNUM)
      BB=-(DTEMP+DNUM)/(DTEMP-DNUM)
  405 CONTINUE
      DO 430 J=1,NFCNS
      FT=(J-1)*DELF
      XT=DCOS(PI2*FT)
      IF(KKK.EQ.1) GO TO 410
      XT=(XT-BB)/AA
      FT=ARCOS(XT)/PI2
  410 XE=X(L)
      IF(XT.GT.XE) GO TO 420
      IF((XE-XT).LT.FSH) GO TO 415
      L=L+1
      GO TO 410
  415 A(J)=Y(L)
      GO TO 425
  420 IF((XT-XE).LT.FSH) GO TO 415
      GRID(1)=FT
      A(J)=GEE(1,NZ)
  425 CONTINUE
      IF(L.GT.1) L=L-1
  430 CONTINUE
      GRID(1)=GTEMP
      DDEN=PI2/CN
      DO 510 J=1,NFCNS
      DTEMP=0.0
      DNUM=(J-1)*DDEN
      IF(NM1.LT.1) GO TO 505
      DO 500 K=1,NM1
  500 DTEMP=DTEMP+A(K+1)*DCOS(DNUM*K)
  505 DTEMP=2.0*DTEMP+A(1)
  510 ALPHA(J)=DTEMP
      DO 550 J=2,NFCNS
  550 ALPHA(J)=2*ALPHA(J)/CN
      ALPHA(1)=ALPHA(1)/CN
      IF(KKK.EQ.1) GO TO 545
      P(1)=2.0*ALPHA(NFCNS)*BB+ALPHA(NM1)
      P(2)=2.0*AA*ALPHA(NFCNS)
      Q(1)=ALPHA(NFCNS-2)-ALPHA(NFCNS)
      DO 540 J=2,NM1
      IF(J.LT.NM1) GO TO 515
      AA=0.5*AA
      BB=0.5*BB
  515 CONTINUE
      P(J+1)=0.0
      DO 520 K=1,J
      A(K)=P(K)
  520 P(K)=2.0*BB*A(K)
      P(2)=P(2)+A(1)*2.0*AA
      JM1=J-1
      DO 525 K=1,JM1
  525 P(K)=P(K)+Q(K)+AA*A(K+1)
      JP1=J+1
      DO 530 K=3,JP1
  530 P(K)=P(K)+AA*A(K-1)
      IF(J.EQ.NM1) GO TO 540
      DO 535 K=1,J
  535 Q(K)=-A(K)
      Q(1)=Q(1)+ALPHA(NFCNS-1-J)
  540 CONTINUE
      DO 543 J=1,NFCNS
  543 ALPHA(J)=P(J)
  545 CONTINUE
      IF(NFCNS.GT.3) RETURN
      ALPHA(NFCNS+1)=0.0
      ALPHA(NFCNS+2)=0.0
      RETURN
      END


      DOUBLE PRECISION FUNCTION D(K,N,M)
C
C  FUNCTION TO CALCULATE THE LAGRANGE INTERPOLATION
C  COEFFICIENTS FOR USE IN THE FUNCTION GEE.
C
      COMMON PI2,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
      DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
      DIMENSION DES(1045),GRID(1045),WT(1045)
      DOUBLE PRECISION AD,DEV,X,Y
      DOUBLE PRECISION Q
      DOUBLE PRECISION PI2
      D=1.0
      Q=X(K)
      DO 3 L=1,M
      DO 2 J=L,N,M
      IF(J-K)1,2,1
    1 D=2.0*D*(Q-X(J))
    2 CONTINUE
    3 CONTINUE
      D=1.0/D
      RETURN
      END


      DOUBLE PRECISION FUNCTION GEE(K,N)
C
C  FUNCTION TO EVALUATE THE FREQUENCY RESPONSE USING THE
C  LAGRANGE INTERPOLATION FORMULA IN THE BARYCENTRIC FORM
C
      COMMON PI2,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
      DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
      DIMENSION DES(1045),GRID(1045),WT(1045)
      DOUBLE PRECISION P,C,D,XF
      DOUBLE PRECISION PI2
      DOUBLE PRECISION AD,DEV,X,Y
      P=0.0
      XF=GRID(K)
      XF=DCOS(PI2*XF)
      D=0.0
      DO 1 J=1,N
      C=XF-X(J)
      C=AD(J)/C
      D=D+C
    1 P=P+C*Y(J)
      GEE=P/D
      RETURN
      END


      SUBROUTINE OUCH
      PRINT 1
    1 FORMAT(' ************** FAILURE TO CONVERGE ***********'/
     1'0PROBABLE CAUSE IS MACHINE ROUNDING ERROR'/
     2'0THE IMPULSE RESPONSE MAY BE CORRECT'/
     3'0CHECK WITH A FREQUENCY RESPONSE')
      RETURN
      END
```

References

[1] J. H. McClellan and T. W. Parks, "A unified approach to the design of optimum FIR linear phase digital filters," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 697–701, Nov. 1973.

[2] E. W. Cheney, *Introduction to Approximation Theory*. New York: McGraw-Hill, 1966, pp. 72–100.

[3] E. Ya. Remez, "General computational methods of Tchebycheff approximation," Kiev, 1957 (Atomic Energy Translation 4491, pp. 1–85).

[4] B. Gold and K. L. Jordan, "A direct search procedure for designing finite duration impulse response filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 33–36, Mar. 1969.

[5] L. R. Rabiner, B. Gold, and C. McGonegal, "An approach to the approximation problem for nonrecursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 83–106, June 1970.

[6] L. R. Rabiner and K. Steiglitz, "The design of wideband recursive and nonrecursive digital differentiators," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 204–209, June 1970.

[7] O. Herrmann, "Design of nonrecursive digital filters with linear phase," *Electron. Lett.*, pp. 328–329, 1970.

[8] E. Hofstetter, A. V. Oppenheim, and J. Siegel, "A new technique for the design of nonrecursive digital filters," in *Proc. 5th Annu. Princeton Conf. Inform. Sci. and Syst.*, Mar. 1971, pp. 64–72.

[9] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 189–194, Mar. 1972.

[10] L. R. Rabiner, "The design of finite impulse response

digital filters using linear programming techniques," *Bell Syst. Tech. J.*, vol. 51, pp. 1177-1198, July-Aug. 1972.
[11] T. W. Parks and J. H. McClellan, "A program for the design of linear phase finite impulse response digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 195-199, Aug. 1972.
[12] O. Herrmann, "Transversal filters for the Hilbert Transformation," *Arch. Elek. Übertragung.*, vol. 23, pp. 581-587, 1969.
[13] J. H. McClellan, "On the design of one-dimensional and two-dimensional FIR digital filters," Ph.D. dissertation, Rice Univ., Houston, Tex., Apr. 1973.

Computer Recognition of the Continuant Phonemes in Connected English Speech

RUSSELL J. NIEDERJOHN, Member, IEEE, and
IAN B. THOMAS, Member, IEEE

Abstract—A method of phoneme recognition of connected speech is described. Input to the system is assumed to consist of the 24 continuant phonemes in connected English speech. The system first categorizes each successive 20-ms segment of the input speech utterance as either voiced fricative, voiced nonfricative, unvoiced fricative or no-speech, utilizing a measure of the relative energy balance between low and high frequencies. Next, the recognition of each 20-ms segment is performed from a distribution of axis-crossing intervals of speech prefiltered to emphasize each formant frequency range. Segmentation is performed from the results of the recognition of each 20-ms segment and from changes in categorization. Finally, the results of the recognition of each 20-ms segment between each pair of segmentation boundaries are combined and the phonemic sound occurring most frequently is printed out. The system has been trained for a single male speaker. Preliminary results for this speaker and for four 3-4-s sentences indicate: a correct categorization decision for about 97 percent of the input 20-ms segments, a correct recognition for about 78 percent of the input 20-ms segments, and an overall correct phoneme recognition for about 87 percent of the input phonemes.

I. Introduction

Phoneme recognition of speech by machine has been a subject of increasing interest in recent years. As a result, numerous techniques have been developed and applied [1]-[14]. In these techniques, the difficulties associated with achieving phoneme recognition in total generality have forced the employment of constraints on the input speech utterance acceptable by recognition systems. Such constraints include a limitation on the size of the vocabulary (number of phonemes), a limitation on the "naturalness" of the utterance and a limitation on the number of speakers acceptable by the system. The employment of these three constraints, with varying degrees of restriction, has been universal in phoneme recognition systems.

In the system described in this paper, the input speech utterance is constrained to consist of the continuant phonemes in connected English speech. Hence, 24 of the possible 40 or so phonemes of English are acceptable to the recognizer. The system recognizes: the eleven vowels, /i, I, ϵ, æ, Λ, a, ɔ, u, U, o, ɚ/; the four voiced fricatives, /v, ð, z, ʒ/; the four unvoiced fricatives, /f, θ, s, ʃ/; the three nasals /m, n, ŋ/; the two semivowels, /l, r/, and the null phoneme (no speech). It does not presently recognize: the vowel glides, /e, aU, aI, ɔI, iU/; the consonant glides, /j, ω/; the affricatives, /tʃ, d3/; the stop consonants, /b, d, g, p, t, k/; or the glottal fricative /h/. The group of phonemes to be recognized was chosen primarily as a result of the high accuracy achieved in an initial study when recognizing these same phonemes uttered in isolation [14]. It was of interest to determine if this high accuracy of recognition could be accomplished for this same group of phonemes in continuous speech. The resulting recognition system is one that vocabulary restrictions can be lessened as methods of recognizing the remaining phonemes are developed and applied. The constraint on the "naturalness" of the spoken utterance acceptable to the system is not made. It is assumed that no attempt is made to enhance recognition by other than "normal" enunciation or ideal noise conditions. Finally, the system as implemented is "trained" to accept speech from one talker. A suitable training procedure is therefore required prior to recognition.

Four sentences containing 107 phonemes were used as a test of the recognition system. The system responded correctly for about 87 percent of the phonemes. It responded incorrectly for about 4.5 percent and failed to respond for about 8.5 percent of the phonemes.

Manuscript received March 3, 1973; revised June 20, 1973. This paper was partially presented at the 1972 International Conference on Speech Communication and Processing, Boston, Mass.
R. J. Niederjohn is with the Department of Electical Engineering, Marquette University, Milwaukee, Wis. 53233.
I. B. Thomas is with the Department of Electrical Engineering, University of Massachusetts, Amherst, Mass. 01002.